**SOURCE CODE**

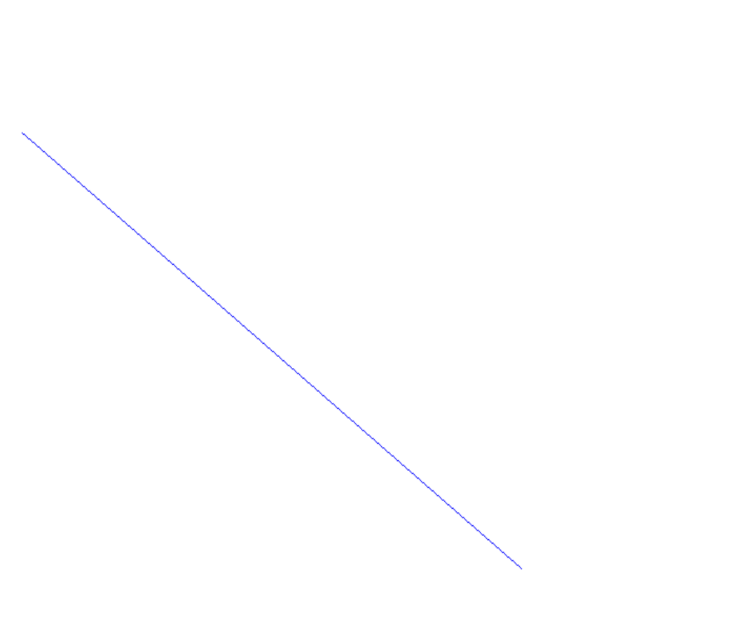
1. #implementation of BLA algorithm in Python
2. **import** pygame
3. **import** sys

6. **def** BLA(x1, y1, x2, y2):
8. dx **=** abs(x2 **-** x1)
9. dy **=** abs(y2 **-** y1)
11. x **=** x1
12. y **=** y1
14. **if**(x2> x1):
15. lx **=** 1
16. **else**:
17. lx **=** **-**1
19. **if**(y2 > y1):
20. ly **=** 1
21. **else**:
22. ly **=** **-**1
24. **if** dx > dy:
25. p **=** 2**\***dy**-**dx
27. **for** i **in** range(dx**+**1):
28. **if** p<0:
29. x **=** x**+**lx
30. y **=** y
31. p **=** p **+** 2**\***dy
32. **else**:
33. x **=** x **+** lx
34. y **=** y **+** ly
35. p **=** p **+** 2**\***(dy **-** dx)
37. screen.set\_at((round(x), round(y)), WHITE)#plot(int(x), int(y))
38. **else**:
39. p **=** 2**\***dx**-**dy
41. **for** i **in** range(dy**+**1):
42. **if** p<0:
43. x **=** x
44. y **=** y **+** ly
45. p **=** p **+** 2**\***dx
46. **else**:
47. x **=** x **+** lx
48. y **=** y **+** ly
49. p **=** p **+** 2**\***(dx **-** dy)
51. screen.set\_at((round(x), round(y)), WHITE)

54. pygame.init()
55. WIDTH **=** 800
56. HEIGHT **=** 600
58. screen **=** pygame.display.set\_mode((WIDTH, HEIGHT))
60. pygame.display.set\_caption("Bresenham's Line Algorithm")
62. WHITE **=** (12, 15, 255)
63. BLACK **=** (255, 255, 255)

66. **def** main():
67. **while** True:
68. **for** event **in** pygame.event.get():
69. **if** event.type**==**pygame.QUIT:
70. pygame.quit()
71. sys.exit()
72. screen.fill(BLACK)
73. BLA(25, 125, 200, 220)
74. pygame.display.flip()
75. pygame.time.delay(100)
77. **if** \_\_name\_\_ **==** "\_\_main\_\_":
78. main()

**OUTPUT**



*Fig. 3.1: Output of Bresenham's Line Algorithm (BLA)*